UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/666,086	09/19/2003	James T. Pinkerton	13768.421.2	8991

47973 7590 10/22/2007

WORKMAN NYDEGGER/MICROSOFT
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UT 84111

EXAMINER	
ALAM, UZMA	

ART UNIT	PAPER NUMBER
2157	

MAIL DATE	DELIVERY MODE
10/22/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

mV

Office Action Summary	Application No.	Applicant(s)	
	10/666,086	PINKERTON ET AL.	
	Examiner	Art Unit	
	Uzma Alam	2157	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *03 August 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-53* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-53* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *19 September 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

This action is responsive to the amendments filed August 3, 2007. Claims 1-53 are pending. Claims 1-53 represent a method for offloading state objects.

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1-5, 8-14, 16, 21-38, 40-43, 45-48, 50-53 are rejected under 35 U.S.C. 103(a) as being unpatentable over Boucher et al. US Patent No. 6,334, 153 in view of Anand et al. US Patent No. 6,370,599. Anand et al. teaches the invention as claimed including transferring functions from one peripheral device to another based on the resource availability of the device (see abstract). Boucher and Anand teach the invention as claimed including a system for protocol processing (see abstract).

3.

As per claim 1, Boucher teaches in a computerized system comprising a switching layer, and a sequence of one or more intermediate software layers of a network protocol stack, each of the intermediate software layers having a state object, a method for transferring control between one or more destination component devices and one or more source component devices, the

control needed to process a plurality of state objects while still maintaining integrity of established network communication, the method comprising the following:

an act of generating an offload data structure, the offload data structure comprising a hierarchy of a plurality of state objects, the hierarchy corresponding to a plurality of connections sharing a common path state object, the plurality of state objects corresponding to a network protocol state for one or more intermediate software layers (the CCB is the data structure; column 5, lines 50-55; column 6, lines 22-65; column 8, lines 41-56; column 9, lines 34-61; short-lived column 11, lines 11-35);

an act of transferring from a source component device to a destination component device two or more state objects in the same intermediate software layer of the offload data structure (sending the object from the source to work it's way through the destination, which is the INIC using the CCB; column 10, lines 28-51); and

an act of the destination component processing the two or more state objects at the same protocol layer after the transfer (the INIC processing the transferred objects; column 3, lines 41-67; column 5, lines 41-67).

Boucher does not teach concurrently transferring two or more state objects. Anand teaches concurrently transferring state objects. See column 4, lines 34-67; column 13, lines 45-67; column 14, lines 1-55; column 15, lines 1-30. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the offloading of Boucher with the concurrent offloading of Anand. One of ordinary skill in the art would have been motivated to do this to increase computing efficiency (Anand; column 15, lines 5-25)

As per claim 2, Boucher and Anand teach the method of claim 1, wherein one of the destination component and source component is a peripheral device (Boucher; the INIC; column 5, lines 6-16; column 9, lines 30-35; column 17, lines 26-37).

As per claim 3, Boucher and Anand teach the method of claim 2, wherein at least one of the two or more state objects includes a cached state variable (Boucher: the CCB and other cached blocks; column 5, lines 34-41; column 7, lines 21-30; column 8, lines 1-5; constant and delegated state variables; column 5, lines 41-54).

As per claim 4, Boucher and Anand teach the method of claim 2, wherein the offload data structure includes a block list having at least a next block pointer that points to the same intermediate software layer of a different connection path through the network protocol stack, and a dependent block pointer that points to a different hierarchal layer of the same connection path through the network protocol stack (Boucher: the CCB compresses multiple layers into one block to be able to process though the intermediate layers; column 7, lines 21-45).

As per claim 5, Boucher and Anand teach the method of claim 2, wherein the act of transferring the offload data structure and the portion of the state object for each software layer is performed during the slow-start phase of a TCP connection (Boucher: column 7, lines 48-60).

As per claim 8, Boucher and Anand teach the method of claim 2, further comprising:

an act of the peripheral device generating one or more handles for one or more intermediate software layers, wherein the peripheral device transfers the one or more handles to one or more corresponding host protocol stack layers (Boucher: column 11, lines 1-51); and

an act of, if at least one of the intermediate software layers changes cached state, the at least one of the intermediate software layers updating the cached state of the peripheral device (Boucher: column 12, lines 26-41).

As per claim 9, Boucher and Anand teach the method of claim 2, wherein a failover event occurs, further comprising:

an act of detecting that one or more links being processed at one or more corresponding peripheral devices have failed, the one or more failed peripheral devices having been given processor control of one or more offload data structures and one or more state objects (transferring control to the host of the source cannot process the object; column 5, lines 56-67).

Boucher does not teach an act of detecting a different one or more peripheral devices that are capable of handling processing control of the one or more links by receiving the one or more offload data structures and control of the one or more state objects.

Anand teaches an act of detecting a different one or more peripheral devices that are capable of handling processing control of the one or more links by receiving the one or more offload data structures and control of the one or more state objects. See column 13, lines 30-44. It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine the additional peripheral devices of Anand with the INIC of Boucher. A person of

ordinary skill in the art would have been motivated to do this to add redundant resources to the already present resources of Boucher.

As per claim 10, Boucher and Anand teach the method of claim 9, further comprising transferring to the source component the one or more offload data structures and processing control of the one or more state objects from the one or more failed peripheral devices, thereby uploading processing control of the one or more links from the one or more failed peripheral devices to the source component (Boucher; sending the object from the INIC back to the source; Boucher column 5, lines 56-67).

As per claim 11, Boucher and Anand teach the method of claim 10, further comprising an act of detecting a processing resource of the different one or more peripheral devices; and adjusting the one or more state objects that have been uploaded from the one or more failed peripheral devices to match the detected resource of the different one or more peripheral devices, so that the one or more links can be processed at the different one or more peripheral devices rather than at the failed one or more peripheral devices (Boucher updating the CCB; column 5, lines 56-67).

As per claim 12, Boucher and Anand teach the method of claim 11, wherein the one or more links are aggregated 802.3ad links (column 10, line 45, column 16, lines 16-39).

As per claim 13, Boucher and Anand teach the method of claim 9, further comprising:

an act of detecting a processing resource of the different one or more peripheral devices (Boucher; column 5, lines 55-67);

if one of the one or more offloaded state objects can be processed by the different one or more peripheral devices based on the detected processing resource, transferring processing control of the one of the one or more offloaded state objects to the different one or more peripheral devices (Boucher; column 6, lines 1-21); and

if the one or more offloaded state objects cannot be processed by the different one or more peripheral devices based on the detected processing resource, transferring processing control of the one of the one or more offloaded state objects to the host protocol stack (Boucher; column 5, lines 64-67).

As per claim 14, Boucher and Anand teach the method of claim 13, wherein the one or more links are aggregated 802.3ad links (Boucher; column 10, line 45, column 16, lines 16-39).

As per claim 16, Boucher and Anand teach the method of claim 15, wherein the destination component is a peripheral device (Boucher; column 6, lines 1-21).

As per claim 35, Boucher and Anand teach the method of claim 1 wherein the computerized system comprises a central processing unit, and a plurality of peripheral devices, wherein the transfer of control corresponds to transferring a plurality of offloaded state objects

from one or more of the peripheral devices while still maintaining integrity off established network communication, the method further comprising the following:

an act of detecting that one or more links being processed at one or more peripheral devices has failed, the one or more failed peripheral devices having been given processing control of one or more state objects (transferring control to the host of the source cannot process the object; column 5, lines 56-67);

an act of transferring processing control to the one or more detected different one or more different peripheral devices (column 6, lines 1-21).

Boucher does not teach an act of detecting a different one or more peripheral devices that are capable of handling processing control of only the one or more links;

an act of detecting a different one or more peripheral devices that are capable of receiving processing control of the one or more links and one or more state objects; and

an act of detecting a processing resource of any of the detected different one or more peripheral devices.

Anand teaches an act of detecting a different one or more peripheral devices that are capable of handling processing control of only the one or more links;

an act of detecting a different one or more peripheral devices that are capable of receiving processing control of the one or more links and one or more state objects; and

an act of detecting a processing resource of any of the detected different one or more peripheral devices.

See column 13, lines 30-44.

It would have been obvious to a person of ordinary skill in the art at the time of the invention to

combine the additional peripheral devices of Anand with the INIC of Boucher. A person of ordinary skill in the art would have been motivated to do this to add redundant resources to the already present resources of Boucher.

As per claim 36, Boucher and Anand teach the method of claim 35, further comprising:

if the different one or more peripheral devices are capable of handling processing control of only the one or more links based on the detected processing resource, an act of transferring processing control of only the one or more links to the different one or more peripheral devices (Boucher; column 6, lines 1-21);

if the different one or more peripheral devices are capable of handling processing control of one or more links and one or more state objects based on the detected processing resource, an act of transferring processing control of the one or more links and the one or more offloaded state objects to the different one or more peripheral devices (Boucher; column 5, lines 64-67); and

an act of transferring processing control of any remaining of the one or more links and any remaining of the one or more offloaded state objects to the host protocol stack (Boucher; column 5, lines 55-67; column 6, lines 1-21).

As per claim 37, Boucher and Anand teach the method of claim 36, wherein one or more of the offloaded state objects include at least a cached state variable (Boucher; the CCB; column 5, lines 34-41; column 7, lines 21-30; column 8, lines 1-5).

As per claim 38, Boucher and Anand teach the method of claim 36, wherein the one or more links are aggregated 802.3ad links (Boucher; column 10, line 45; column 16, lines 16-39.

As per claim 40, Boucher and Anand teach the method of claim 1, wherein the system further comprises a host processing unit, and one or more peripheral devices, and wherein at least some of the intermediate software layers have a state object, and wherein the method further comprises:

an act of inserting one or more intermediate driver data structures within the offload data structure (Boucher: sending the object from the source to work it's way through the destination, which is the INIC using the CCB; column 10, lines 28-51); and

wherein transferring comprises transferring from the source component device to one or more peripheral devices the two or more state objects (Boucher: the INIC processing the transferred objects; column 3, lines 41-67; column 5, lines 41-67).

As per claim 41, Boucher and Anand teach the method of claim 40, wherein a failover event occurs, an intermediate driver corresponding with at least one of the one or more intermediate driver data structures performs a method comprising:

an act of detecting a different one or more peripheral devices that are capable of handling processing control of only the one or more links (Boucher; column 5, lines 55-67);

an act of detecting a different one or more peripheral devices that are capable of receiving processing control of the one or more links and one or more state objects (Boucher; column 6, lines 1-21); and

an act of detecting a processing resource of any of the detected one or more peripheral devices (Boucher; column 5, lines 64-67).

As per claim 42, Boucher and Anand teach the method of claim 40, wherein the one or more intermediate driver data structures are inserted into the offload data structure prior to offloading processing control of the offload data structure to the one or more peripheral devices (Boucher: the INIC; column 5, lines 6-16; column 9, lines 30-35; column 17, lines 26-37).

As per claim 43, Boucher and Anand teach the method of claim 40, wherein the one or more intermediate driver data structures are inserted at the point of initiating an offload request of the offload data structure to the one or more peripheral devices (Boucher: the INIC; column 5, lines 6-16; column 9, lines 30-35; column 17, lines 26-37).

As per claim 45, Boucher and Anand teach the method of claim 40, wherein the offload data structure corresponds with two or more state objects corresponding to the same intermediate software layer, and wherein the two or more state objects point to a single state object at a lower software layer of a network protocol stack, wherein the offload data structure has the appearance of an inverted tree (Boucher: column 12, lines 61-67; column 13, lines 1-10).

As per claim 46, Boucher and Anand teach the method of claim 45, further comprising an act of transferring processing control of the entire inverted tree offload data structure to one or more of the one or more peripheral devices (Boucher: column 12, lines 61-67; column 13, lines 1-10).

As per claim 47, Boucher and Anand teach the method of claim 46, further comprising an act of transferring processing control of one or more of the state objects of the inverted tree offload data structure from the at least one of the one or more peripheral devices to the host protocol stack (Boucher: column 12, lines 61-67; column 13, lines 1-10).

As per claim 48, Boucher and Anand teach the method of claim 47. Boucher does not teach wherein the act of transferring from a source component device to one or more peripheral devices one or more state objects occurs in response to an act of detecting that at least one of the one or more peripheral devices has failed. Anand teaches wherein the act of transferring from a source component device to one or more peripheral devices one or more state objects occurs in response to an act of detecting that at least one of the one or more peripheral devices has failed. See column 13, lines 30-44.
It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine the additional peripheral devices of Anand with the INIC of Boucher. A person of ordinary skill in the art would have been motivated to do this to add redundant resources to the already present resources of Boucher.

As per claim 49, Boucher and Anand teach the method of claim 48, wherein an intermediate driver that corresponds with at least one of the one or more intermediate driver data structures performs the method comprising:

an act of detecting a different one or more peripheral devices that are capable of handling processing control of only the one or more links (Boucher updating the CCB; column 5, lines 56-67);

an act of detecting a different one or more peripheral devices that are capable of receiving processing control of the one or more links and one or more state objects (Boucher updating the CCB; column 5, lines 56-67); and

an act of detecting a processing resource of any of the detected one or more peripheral devices (Boucher updating the CCB; column 5, lines 56-67).

if the different one or more peripheral devices are capable of handling processing control of only the one or more links based on the detected processing resource, an act of transferring processing control of only the one or more links to the different one or more peripheral devices (Boucher; column 5, lines 55-67);

if the different one or more peripheral devices are capable of handling processing control of one or more links and one or more state objects based on the detected processing resource, an act of transferring processing control of the one or more links and the one or more offloaded state objects to the different one or more peripheral devices (Boucher; column 6, lines 1-21); and

an act of transferring processing control of any remaining of the one or more links and any remaining of the one or more offloaded state objects to the host protocol stack (Boucher; column 5, lines 64-67).

As per claim 50, Boucher and Anand teach a computer program product comprising one or more computer-readable storage media storage computer-executable instructions that, when executed by one or more processors at the host computing system, cause the host computing system to perform the method recited in claim 1. See claim 1.

As per claim 51, Boucher and Anand teach the computer program product of claim 50, wherein the computer-executable instructions, when executed by the one or more processors, further cause the computerized system to perform the following:

an act of generating a handle for each layer of the intermediate software layers, wherein the destination component is a NIC, and the NIC transfers the handle for each intermediate software layer to each intermediate software layer (Boucher: column 11, lines 1-51); and

an act of, if at least one of the intermediate software layers changes cached state, the at least one of the intermediate software layers updating the cached state of the NIC (Boucher: column 12, lines 26-41).

As per claim 52, Boucher and Anand teach the computer program product of claim 50, wherein the computer-executable instructions, when executed by the one or more processors, further cause the computerized system to perform the following an act of detecting that one or

more links being processed at one or more corresponding peripheral devices have failed, the one or more failed peripheral devices having been given processor control of one or more offload data structures and one or more state objects (transferring control to the host of the source cannot process the object; column 5, lines 56-67).

Boucher does not teach an act of detecting a different one or more peripheral devices that are capable of handling processing control of the one or more links by receiving the one or more offload data structures and control of the one or more state objects.

Anand teaches an act of detecting a different one or more peripheral devices that are capable of handling processing control of the one or more links by receiving the one or more offload data structures and control of the one or more state objects. See column 13, lines 30-44. It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine the additional peripheral devices of Anand with the INIC of Boucher. A person of ordinary skill in the art would have been motivated to do this to add redundant resources to the already present resources of Boucher.

As per claim 53, Boucher and Anand teach the computer program product of claim 52, wherein the computer-executable instructions, when executed by the one or more processors, further cause the computerized system to perform the following:

an act of detecting a processing resource of the different one or more peripheral devices (Boucher; column 5, lines 55-67);

if one of the one or more offloaded state objects can be processed by the different one or

more peripheral devices based on the detected processing resource, transferring processing control of the one of the one or more offloaded state objects to the different one or more peripheral devices (Boucher; column 6, lines 1-21); and

if the one or more offloaded state objects cannot be processed by the different one or more peripheral devices based on the detected processing resource, transferring processing control of the one of the one or more offloaded state objects to the host protocol stack (Boucher; column 5, lines 64-67).

4. Claims 6, 7, 15, 17-20, 39 and 44 is rejected under 35 U.S.C. 103(a) as being unpatentable over Boucher et al. US Patent No. 6,334, 153 in view of Anand et al. US Patent No. 6,370,599 as applied to claims 35-38 above, and further in view of Dwork et al. US Patent No. 6,963,946. Dwork et al. teaches the invention as claimed including transferring functions to a peripheral device (see abstract).

As per claim 6, Boucher and Anand teach the method of claim 2. Boucher does not teach further comprising an act of inserting an intermediate driver data structure in the offload data structure, the intermediate driver capable of deciphering and encapsulating incoming and outgoing data packets, as appropriate, wherein the outgoing data packets are packaged with one or more virtual local area network identifiers, and incoming packets are stripped of one or more virtual local area network identifiers. Dwork teaches an act of inserting an intermediate driver data structure in the offload data structure, the intermediate driver capable of deciphering and

encapsulating incoming and outgoing data packets, as appropriate, wherein the outgoing data packets are packaged with one or more virtual local area network identifiers, and incoming packets are stripped of one or more virtual local area network identifiers. See column 8, lines 41-56. It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine the identifiers and tags of Dwork with the headers of Boucher. A person of ordinary skill in the art would have been motivated to do this to expand the functionalities of the network of Boucher.

As per claim 7, Boucher and Anand teach the method of claim 6. Boucher does not teach wherein the intermediate driver is capable of securing network traffic over one or more network links using an IPSEC protocol by performing one or more of adding and removing an authentication header to a data packet, and encrypting and decrypting the data packet, as appropriate. Dwork teaches wherein the intermediate driver is capable of securing network traffic over one or more network links using an IPSEC protocol by performing one or more of adding and removing an authentication header to a data packet, and encrypting and decrypting the data packet, as appropriate. See column 20, lines 45-67, column 21, lines 1-60. It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine the identifiers and tags of Dwork with the headers of Boucher. A person of ordinary skill in the art would have been motivated to do this to expand the functionalities of the network of Boucher.

As per claim 15, Boucher and Anand teach the method of claim 1. Boucher and Anand do not teach wherein the computerized system processing two or more state objects corresponding to the same intermediate software layer and wherein maintaining integrity of

established network communication comprises preserving the integrity of network communications with regard to state changes, failover events, and VLAN tags. Dwork teaches wherein the computerized system processing two or more state objects corresponding to the same intermediate software layer and wherein maintaining integrity of established network communication comprises preserving the integrity of network communications with regard to state changes, failover events, and VLAN tags. See column 2, lines 30-60.

It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine the identifiers and tags of Dwork with the headers of Boucher. A person of ordinary skill in the art would have been motivated to do this to expand the functionalities of the network of Boucher.

As per claim 17, Boucher and Anand teach the method of claim 16, further comprising a step for updating the destination component in the event of a change in the state object, or in the event of a change in the offload data structure so that the destination component can still process the two or more state objects as intended without losing network communication integrity (Boucher; updating the CCB; column 5, lines 56-67).

As per claim 18, Boucher and Anand teach the method of claim 16, wherein the step for updating the destination component includes the following: an act of the peripheral device generating one or more handles for one or more intermediate software layers, wherein the

peripheral device transfers the one or more handles to one or more corresponding host protocol stack layers (Boucher; column 11, lines 1-51); and

an act of, if at least one of the intermediate software layers changes cached state, the at least one of the intermediate software layers updating the cached state of the peripheral device (Boucher; column 12, lines 26-41).

5. As per claim 19, Boucher and Anand teach the method of claim 16, further comprising a step for dynamically preserving the integrity of the network communications in the event of a failover event so that the one or more aggregated network links can be processed appropriately at a different one or more peripheral devices (Boucher; sending the object from the INIC back to the source; Boucher column 5, lines 56-67).

As per claim 20, Boucher, Anand and Dwork teach the method of claim 19, wherein the step for dynamically preserving the integrity of the two or more network links includes:

an act of detecting that one or more links being processed at one or more corresponding peripheral devices have failed, the one or more failed peripheral devices having been given processor control of one or more offload data structures and one or more state objects (Boucher: transferring control to the host of the source cannot process the object; column 5, lines 56-67).

Boucher does not teach an act of detecting a different one or more peripheral devices that are capable of handling processing control of the one or more links by receiving the one or more offload data structures and control of the one or more state objects;

if one of the one or more offloaded state objects can be processed by the different one or more peripheral devices based on the detected processing resource, transferring processing control of the one of the one or more offloaded state objects to the different one or more peripheral devices; and if the one or more offloaded state objects cannot be processed by the different one or more peripheral devices based on the detected processing resource, transferring processing control of the one of the one or more offloaded state objects to the host protocol stack.

Anand teaches an act of detecting a different one or more peripheral devices that are capable of handling processing control of the one or more links by receiving the one or more offload data structures and control of the one or more state objects; if one of the one or more offloaded state objects can be processed by the different one or more peripheral devices based on the detected processing resource, transferring processing control of the one of the one or more offloaded state objects to the different one or more peripheral devices; and if the one or more offloaded state objects cannot be processed by the different one or more peripheral devices based on the detected processing resource, transferring processing control of the one of the one or more offloaded state objects to the host protocol stack.. See column 13, lines 30-44.

It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine the additional peripheral devices of Anand with the INIC of Boucher. A person of ordinary skill in the art would have been motivated to do this to add redundant resources to the already present resources of Boucher.

As per claim 39, Boucher and Anand teach the method of claim 36. Boucher and Anand do not teach further comprising, prior to the act of transferring processing control of the one or more links and the one or more offloaded state objects, inserting an intermediate driver adjacent a data structure comprising the offloaded state objects, the intermediate driver capable of performing at least one of multiplexing one or more VLAN tags, de-multiplexing one or more VLAN tags, directing network state object traffic over one or more peripheral devices, and securing network data packets using an IPSEC protocol. Boucher does not teach a step for reducing the processing demands of a computerized system that is processing two or more state objects corresponding to the same intermediate software layer by transferring processing control of the two or more state objects in a way that preserves the integrity of network communications with regard to state changes, failover events, and VLAN tags.

Dwork teaches further comprising, prior to the act of transferring processing control of the one or more links and the one or more offloaded state objects, inserting an intermediate driver adjacent a data structure comprising the offloaded state objects, the intermediate driver capable of performing at least one of multiplexing one or more VLAN tags, de-multiplexing one or more VLAN tags, directing network state object traffic over one or more peripheral devices, and securing network data packets using an IPSEC protocol. Boucher does not teach a step for reducing the processing demands of a computerized system that is processing two or more state objects corresponding to the same intermediate software layer by transferring processing control of the two or more state objects in a way that preserves the integrity of network communications with regard to state changes, failover events, and VLAN tags. See column 2, lines 30-60.

It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine the identifiers and tags of Dwork with the headers of Boucher. A person of ordinary skill in the art would have been motivated to do this to expand the functionalities of the network of Boucher.

As per claim 44, Boucher and Anand teach the method of claim 43. Boucher does not teach wherein, at least one of the one or more intermediate drivers includes instructions for performing an act of directing the sub-data packet to a VLAN address based on a virtual LAN tag associated with the sub-data packet, the instructions further comprising and at least one of: an act of receiving a multiplexed data packet; an act of de-multiplexing the data packet into one or more sub-data packets; an act of multiplexing a data packet; and an act of sending a multiplexed data packet. Dwork teaches wherein, at least one of the one or more intermediate drivers includes instructions for performing an act of directing the sub-data packet to a VLAN address based on a virtual LAN tag associated with the sub-data packet, the instructions further comprising and at least one of: an act of receiving a multiplexed data packet; an act of de-multiplexing the data packet into one or more sub-data packets; an act of multiplexing a data packet; and an act of sending a multiplexed data packet. See column 2, lines 30-60.

It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine the identifiers and tags of Dwork with the headers of Boucher. A person of ordinary skill in the art would have been motivated to do this to expand the functionalities of the network of Boucher.

6. Claims 21-34 are rejected under the same rationale as claims 1-34 because they are directed to the same method as claims 1-14.

Response to Arguments

7. Applicant's arguments with respect to claims 1-53 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Uzma Alam whose telephone number is (571) 272-3995. The examiner can normally be reached on Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on (571) 272-4001. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Uzma Alam
Ua
October 11, 2007

ARIO ETIENNE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100